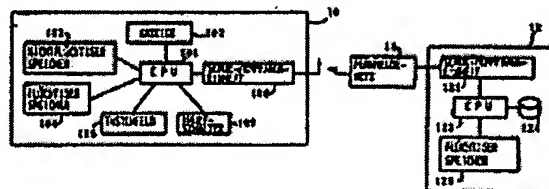


Software updating method for microcomputer-supported mobile telephone

Patent number: DE19543843
Publication date: 1997-05-28
Inventor: CHEN VICTOR (TW)
Applicant: ACER PERIPHERALS INC (TW)
Classification:
- **International:** H04Q3/545; H04M1/00; H04Q7/20
- **European:** G06F9/445E; H04M1/725F1A; H04Q3/545C1; H04Q7/32A2
Application number: DE19951043843 19951124
Priority number(s): DE19951043843 19951124

Abstract of DE19543843

The method is applied to a mobile telephone (10) which is linked by a telecommunications network (11) to a software-updating computer system (12). The computer system includes a transceiver (121), a volatile memory (123) and a hard disc (124). The mobile telephone holds its software in a non-volatile flash memory (103) with a structure table containing its version number. The software is set into an updating mode and a link is established with the remote computer system, to which the structure table is transmitted. An updating package with new operating system and function modules and a modified structure table is then received by the mobile telephone, via its transceiver (106).



Data supplied from the *esp@cenet* database - Worldwide

HOFFMANN · EITLE

MÜNCHEN LONDON

**Translation of the relevant Parts of the German Offenlegungsschrift
DE 195 43 843 A1**

Applicant: Acer Peripherals Inc., Kweishan, Taoyuan, TW
Date of Application: November 24, 1995
Laid-Open Date: May 28, 1997
H · E File: 102 911 / aol

Title: Method for the Actualisation of the Software in a Microcomputer-supported Telephone

...
According to Fig. 1, a computer system 12 is connected to a mobile telephone *via* a telecommunication network 11. The telecommunication network 11 is a mobile telephone network. The mobile telephone 10 contains a central processor unit (CPU) 101, an indicator unit 102 for indicating simple messages to the user of the telephone 10, a non-volatile memory 103, as is *e.g.* known under the designation FLASH EPROM, for storing the software for the central processor unit 101, a volatile memory 104, such as *e.g.* a memory for random access (RAM), for executing the program contained in the software or for the actualisation of the software, a key pad 105 for inputting telephone numbers or control codes by the user, a transmitter/receiver unit 106 for the communication with the telecommunication network 11, as well as a main switch 107 controllable by the central processor unit 101.

The computer system 12 is connected to the telecommunication network 11 for the purpose of software actualisation. It contains a transmitter/receiver unit 121 for the communication with the telecommunication network 11, a central processor unit (CPU) 122 and, connected thereto, a volatile memory 123 and a hard disk 124 for the realization of the actualisation.

Fig. 2 diagrammatically shows the software stored in the non-volatile memory 103 of the mobile telephone 10. The software 20 contains several modules, such as the OS module 21 (operating system module), the function module 22 and a structure table 23. The

function modules 22 contain communication modules for the communication with base stations of the telephone network 11 by means of a protocol defined by a certain mobile telephone network, such as *e.g.* a GSM network, and also special program modules (feature modules) for the execution of certain functions of program items in a certain telephone 10, such as *e.g.* a short announcement, a telephone answering function and such like. The OS module 21 forms a real-time operating system for the control of the operations of all modules 22 and also for the actualisation of the software 20. It can be switched over to an actualisation mode for the actualisation of all modules of the software 20.

The structure table 23 contains the entire structure information of the software 20, such as *e.g.* the version number of the software 20 and of the OS module 21, the version number and special parameters of any function module 22, the hardware structure data of the telephone 10, such as the size of the memory and the type of the central processor unit 101, the keypad 105 and the indicator unit 102. This structure table 23 is transmitted to the computer system 12 for determining, how the software 20 is to be actualised. The information contained in the structure table 23 can be changed in accordance with the type of the actualisation service offered by the computer system 12.

The memory assignment plan according to Fig. 3 shows which way the modules 21 and 22 are stored in the non-volatile memory 103. Each of the modules 21, 22 and 23 is stored in a memory slot 31 of its own of the memory 103, a certain additional storage space for later actualisation being assigned to the respective slot. An address table forming part of the structure table 23 serves for holding the addresses of the predetermined memory slots 31. When an existing module is actualised, a new module is stored in the same memory slot starting with the same initial address. Since each memory slot 31 has additional memory space, a deviation in size of the new module does not present a problem, as long as its size does not exceed the available memory space of the corresponding memory slot 31.

This manner of storing allows that each existing module is actualised separately and, therefore, that a complicated storing order or memory reorganisation problems in the memory 103 are avoided. Correspondingly, the OS module 21 can be relatively simple. Further, for the actualisation of the entire software 20, no great volatile memory 104 is required since only few modules will have to be actualised at a time. For most of the actualisation procedures it will be sufficient if the memory 104 is great enough for accommodating three to five modules. In addition, the time for the module transmission from the computer system 12 to the telephone 10 is short.

Fig. 4 represents an actualisation procedure 40 for the actualisation of the software 20 of the mobile telephone 10 which is triggered by the user of the mobile telephone 10. The actualisation procedure 40 has the following steps:

- Step 401 Setting the OS module 21 of the software in an actualisation mode
- Step 402 Keying-in the telephone number of the computer system 12 for setting-up a telecommunication connection with the computer system 12 *via* the telecommunication network 11; if the network 11 is a GSM network for the use of a data service from a base station of the network 11 in the vicinity, which thereupon transmits a subsequent information in the form of digital data, a data request follows from the OS module 21
- Step 403 Letting the user input an option code for the indication of an actualisation option
- Step 404 Transmitting the structure table 23 and (if input) the user option code to the computer system 12
- Step 405 Upon receipt of the structure table 23 from the telephone, the computer system 12 produces an actualisation packet with new modules and a new structure table and transmits the same to the telephone 10
- Step 406 Storing the actualisation packet in the volatile memory 104
- Step 407 Checking the received actualisation packet
- Step 408 Blocking the main switch 107 in order to prevent an interruption by the user during the following actualisation step (which could destroy both the software 20 and the actualisation packet stored in the volatile memory 104)
- Step 409 Actualising the software 20 by storing the new modules and the new structure table from the actualisation packet in the individual corresponding memory slots 31
- Step 410 Releasing the main switch 107.

Claims

1. A method for the actualisation of the software (20) of a microcomputer-supported telephone (10) comprising a central processor unit 101, a volatile memory (104) for the execution of the program and a non-volatile memory (103) in which the software (20) is stored, the software (20) having several modules (21, 22),

characterized in

that the method is performed by means of a remotely mounted computer system (12) via a telephone communication network (11), the software (20) stored in the non-volatile memory (103) further having a structure table (23) with at least a version number regarding the software (20), the software (20) being switchable from a normal operation mode to an actualisation mode for actualising the software (20), and **that** the method comprises the following steps:

- setting the software (20) in an actualisation mode,
- setting up a telecommunication connection to the computer system (12),
- transmitting the structure table (23) to the computer system (12),
- producing and transmitting an actualisation packet with new modules and a new structure table corresponding to the structure table (23) received by the computer system (12), and
- actualising the existing modules (21, 22) and the structure table (23) of the software (20) by means of the new modules and the new structure table from the actualisation packet.

19 BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENTAMT

12 Offenlegungsschrift
10 DE 195 43 843 A 1

51 Int. Cl. 6:
H04 Q 3/545
H 04 M 1/00
H 04 Q 7/20

21 Aktenzeichen: 195 43 843.4
22 Anmeldetag: 24. 11. 95
43 Offenlegungstag: 28. 5. 97

DE 195 43 843 A 1

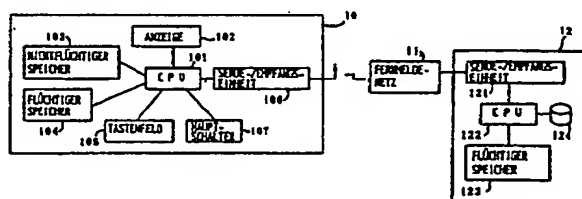
71 Anmelder:
Acer Peripherals Inc., Kweishan, Taoyuan, TW

74 Vertreter:
Patentanwälte Dr. Boeters, Bauer, Dr. Meyer, 81541
München

72 Erfinder:
Chen, Victor, Kweishan, Taoyuan, TW

54 Verfahren zur Aktualisierung der Software in einem mikrocomputergestützten Telefon

- 57 Es wird ein Verfahren zur Aktualisierung der Software eines mikrocomputergestützten Telefons (10), insbesondere Mobiltelefons, mittels eines entfernt angeordneten Computersystems (12) angegeben, das mit dem Telefon (10) über ein Telekommunikationsnetz (11), so z. B. das Mobiltelefonnetz, verbunden ist. Das Verfahren weist die folgenden Verfahrensschritte auf:
- Versetzen der Software (20) des Telefons (10) in einen Aktualisierungsmodus,
 - Herstellen einer Telekommunikationsverbindung mit dem Computersystem (12),
 - Übertragen einer Strukturtafel (23) des Telefons (10) an das Computersystem (12),
 - Erstellen eines Aktualisierungspakets mit neuen Softwaremodulen und einer neuen Strukturtafel durch das Computersystem (12) und Übertragen desselben an das Telefon (10) und
 - Aktualisieren der Software (20) des Telefons (10) mittels des Aktualisierungspakets.



DE 195 43 843 A 1

Die Erfindung betrifft ein Verfahren zur Aktualisierung der Software in einem mikrocomputergestützten Telefon gemäß Gattungsbegriff des Patentanspruchs 1.

Die sich rasch entwickelnde Telefontechnik, vor allem Mobiltelefontechnik, erfordert eine gelegentliche Aktualisierung der in einem derartigen Telefon gespeicherten Software. Viele Mobiltelefonanbieter verlangen von einem Benutzer, daß er zur Aktualisierung der Software sein Mobiltelefon zu einer Kundendienststelle bringt. Dies ist jedoch ziemlich unpraktisch, vor allem wenn sich die Kundendienststelle weit entfernt befindet. Andererseits ist es auch unbequem für die Kundendienststelle, da sie über Techniker und Ausrüstungen für die Durchführung solcher Aktualisierungen verfügen muß.

Zur Lösung dieser Probleme wurden bereits verschiedene Aktualisierungstechniken angegeben. Die US-PS 5,418,837 (Johansson et al.) beschreibt ein Verfahren, welches eine SUM(= Software-Upgrading-Module-)Karte für derartige Aktualisierungen verwendet. Die SUM-Karte wird von einem SIM(= Subscriber-Identity-Module-)Kartenleser gelesen, der sich in einigen Mobiltelefonen, wie zum Beispiel GSM(= Global-System-For-Mobile-Communications-)Terminals, befindet. Obgleich es für den Benutzer damit möglich ist, sein Mobiltelefon mittels der SUM-Karte zu aktualisieren, anstatt dafür eine Kundendienststelle aufsuchen zu müssen, hat diese Technik doch einige Nachteile. Zunächst verlangt die Beschaffung einer korrekten SUM-Karte für seine Bedürfnisse von dem Benutzer Zeit und Mühe. Sodann muß sich der Benutzer, falls die erhaltene SUM-Karte inkorrekt ist oder er seine Optionen für die Aktualisierung seines Mobiltelefons ändert, eine andere SUM-Karte beschaffen, um den Vorgang zu beenden.

Eine andere Technik, wie sie in der US-PS 5,430,877 (Naylor) angegeben ist, gestattet es, mittels eines Mobiltelefons ein oder mehrere über eine unmittelbare Partner-zu-Partner-Verbindung damit verbundene weitere Mobiltelefone zu aktualisieren, sofern es selbst eine Software mit höherer Versionsnummer aufweist. Um dieses Verfahren anwenden zu können, muß der Benutzer aber zunächst einmal ein Mobiltelefon finden, welches eine Software mit höherer Versionsnummer hat. Dies ist sehr unpraktisch, wenn ihm kein entsprechender Telefonteilnehmer bekannt ist. Dazu noch müssen sämtliche verbundenen Mobiltelefone miteinander kompatibel sein, um das Verfahren anwenden zu können.

Auf diesem Hintergrund liegt der Erfindung die Aufgabe zugrunde, ein verbessertes Aktualisierungsverfahren für Telefone gemäß Gattungsbegriff des Anspruchs 1 anzugeben, welches den Benutzer einer Mitwirkung hierzu weitgehend entbindet.

Diese Aufgabe ist erfindungsgemäß durch die Merkmale des Patentanspruchs 1 gelöst. Die Unteransprüche geben darüberhinausgehend vorteilhafte Ausgestaltungsmöglichkeiten an.

Das betreffende Verfahren sieht vor, daß das Telefon zunächst eine Telekommunikationsverbindung mit einem Computersystem herstellt und dem Computersystem eine Strukturtabelle übermittelt. Das Computersystem erstellt daraufhin ein Aktualisierungspaket entsprechend der empfangenen Strukturtabelle und überträgt dieses zurück zu dem Telefon. Das Aktualisierungspaket findet dort dazu Verwendung, die Software des Telefons zu aktualisieren, wobei es zunächst noch zwecks Prüfung in einem Speicher des Telefons zwischengespeichert werden kann.

Nachfolgend wird ein bevorzugtes Ausführungsbeispiel anhand der Zeichnungen genauer beschrieben. Dabei zeigt

Fig. 1 ein Blockschaltbild eines entsprechenden Mobiltelefons und eines über ein Telekommunikationsnetz damit verbundenen Computersystems.

Fig. 2 eine schematische Darstellung der in dem Telefon gespeicherten Software einschließlich einer Strukturtabelle.

Fig. 3 einen Speicherzuordnungsplan, der in dem nicht flüchtigen Speicher des Telefons gespeicherte Softwaremodule zeigt, und

Fig. 4 ein Flußdiagramm, welches den Verfahrensablauf bei der Aktualisierung der Software eines Mobiltelefons wiedergibt.

Gemäß Fig. 1 ist mit einem Mobiltelefon 10 über ein Telekommunikationsnetz 11 ein Computersystem 12 verbunden. Das Telekommunikationsnetz 11 ist ein Mobiltelefonnetz. Das Mobiltelefon 10 enthält eine Zentralprozessoreinheit (CPU) 101, eine Anzeigeeinheit 102 zur Anzeige einfacher Mitteilungen an den Benutzer des Telefons 10, einen nichtflüchtigen Speicher 103, wie er beispielsweise unter der Bezeichnung FLASH EPROM bekannt ist, zur Softwarespeicherung für die Zentralprozessoreinheit 101, einen flüchtigen Speicher 104, wie zum Beispiel einen Speicher für wahlfreien Zugriff (RAM), für die Ausführung des in der Software enthaltenen Programms oder für die Aktualisierung der Software, ein Tastenfeld 105 zur Eingabe von Telefonnummern oder Steuer codes seitens des Benutzers, eine Sende-/Empfangs-Einheit 106 für die Kommunikation mit dem Telekommunikationsnetz 11 sowie einen von der Zentralprozessoreinheit 101 steuerbaren Hauptschalter 107.

Das Computersystem 12 ist zwecks Softwareaktualisierung an das Telekommunikationsnetz 11 angeschlossen. Es enthält eine Sende-/Empfangs-Einheit 121 für die Kommunikation mit dem Netz 11, eine Zentralprozessoreinheit (CPU) 122 sowie, damit verbunden, einen flüchtigen Speicher 123 und eine Festplatte 124 zur Abwicklung des Aktualisierungsvorganges.

Fig. 2 zeigt schematisch die in dem nichtflüchtigen Speicher 103 des Mobiltelefons 10 gespeicherte Software 20. Die Software 20 enthält mehrere Module wie zum Beispiel den OS(= Operating-System-)Modul 21, Funktionsmodule 22 und eine Strukturtabelle 23. Die Funktionsmodule 22 enthalten Kommunikationsmodule für die Kommunikation mit Basisstationen des Telefonnetzes 11 mittels eines von einem bestimmten Mobiltelefonnetz, wie zum Beispiel einem GSM-Netz, definierten Protokolls und ebenso spezielle Programmodule (feature moduls) für die Durchführung bestimmter Funktionen oder Programmpunkte an einem bestimmten Telefon 10, wie zum Beispiel eine Kurzansage, eine Anrufbeantwortungsfunktion und dergl. Der OS-Modul 21 bildet ein Realzeit-Betriebssystem für die Steuerung der Operationen aller Module 22 und ebenso für die Aktualisierung der Software 20. Er kann in einen Aktualisierungsmodus umgeschaltet werden für die Aktualisierung sämtlicher Module der Software 20.

Die Strukturtabelle 23 enthält die gesamte Strukturinformation der Software 20, wie zum Beispiel die Versionsnummer der Software 20 und des OS-Moduls 21, die Versionsnummer und spezielle Parameter eines jeden Funktionsmoduls 22, die Hardwarestrukturdaten des Telefons 10, wie zum Beispiel die Speichergröße und die Art der Zentralprozessoreinheit 101, des Tastenfelds 105 und der Anzeigeeinheit 102. Diese Strukturtabelle

23 wird an das Computersystem 12 übertragen zur Festlegung dessen, wie die Software 20 zu aktualisieren ist. Die in der Strukturtabelle 23 enthaltene Informationen können je nach der Art des von dem Computersystem 12 gebotenen Aktualisierungsdienstes verändert werden.

Der Speicherzuordnungsplan nach Fig. 3 zeigt, in welcher Weise die Module 21 und 22 in dem nichtflüchtigen Speicher 103 gespeichert sind. Jeder der Module 21, 22 und 23 ist in einem eigenen Speicherschlitze 31 des Speichers 103 gespeichert, wobei ihm jeweils ein gewisser zusätzlicher Speicherraum für die spätere Aktualisierung zugeordnet ist. Eine Adressentabelle, die einen Teil der Strukturtabelle 23 bildet, dient dazu, die Adressen der vorbestimmten Speicherschlitze 31 festzuhalten. Bei der Aktualisierung eines vorhandenen Moduls wird in dem gleichen Speicherschlitze ein neuer Modul abgespeichert, wobei mit derselben Anfangsadresse begonnen wird. Da ein jeder Speicherschlitze 31 zusätzlichen Speicherraum aufweist, bildet eine Größenabweichung des neuen Moduls kein Problem, solange dessen Größe nicht über den verfügbaren Speicherraum des betreffenden Speicherschlitzes 31 hinausgeht.

Diese Speicherweise gestattet es, einen jeden vorhandenen Modul getrennt zu aktualisieren, und vermeidet so eine komplizierte Speicherordnung oder Speicherumordnungsprobleme in dem Speicher 103. Entsprechend kann der OS-Modul 21 verhältnismäßig einfach sein. Des weiteren wird für die Aktualisierung der gesamten Software 20 kein großer flüchtiger Speicher 104 benötigt, da ja jeweils nur wenige Module zu aktualisieren sein werden. Für die meisten Aktualisierungsvorgänge wird es genügen, wenn der Speicher 104 groß genug ist, drei bis fünf Module aufzunehmen. Daneben ist die Zeit für die Modulübertragung vom Computersystem 12 zu dem Telefon 10 gering.

Fig. 4 gibt einen Aktualisierungsvorgang 40 für die Aktualisierung der Software 20 des Mobiltelefons 10 wieder, der vom Benutzer des Telefons 10 ausgelöst wird. Der Aktualisierungsvorgang 40 weist folgende Schritte auf:

Schritt 401 Versetzen des OS-Moduls 21 der Software 20 in einen Aktualisierungsmodus

Schritt 402 Eintasten der Telefonnummer des Computersystems 12 zur Herstellung einer Telekommunikationsverbindung mit dem Computersystem 12 über das Telekommunikationsnetz 11; ist das Netz 11 ein GSM-Netz für die Inanspruchnahme eines Datendienstes seitens einer nahegelegenen Basisstation des Netzes 11, die daraufhin zwischen Telefon 10 und Computersystem 12 eine Folgeinformation in Gestalt digitaler Daten überträgt, so erfolgt seitens des OS-Moduls 21 ein Datenabruf

Schritt 403 den Benutzer einen Optionscode zur Angabe einer Aktualisierungsoption eingeben lassen

Schritt 404 Übertragen der Strukturtabelle 23 und (sofern eingegeben) des Benutzeroptionscodes an das Computersystem 12

Schritt 405 auf den Empfang der Strukturtabelle 23 aus dem Telefon 10 hin erstellt das Computersystem 12 ein Aktualisierungspaket mit neuen Modulen und einer neuen Strukturtabelle und überträgt diese an das Telefon 10

Schritt 406 Speichern des Aktualisierungspakets in dem flüchtigen Speicher 104

Schritt 407 Prüfen des empfangenen Aktualisierungspakets

Schritt 408 Sperren des Hauptschalters 107 zur Verhin-

derung einer Unterbrechung durch den Benutzer während des folgenden Aktualisierungsschrittes (die dabei sowohl die Software 20 als auch das in dem flüchtigen Speicher 104 gespeicherte Aktualisierungspaket zerstören könnte)

Schritt 409 Aktualisieren der Software 20 durch Abspeichern der neuen Module und der neuen Strukturtabelle aus dem Aktualisierungspaket in den einzelnen entsprechenden Speicherschlitzen 31

Schritt 410 Freigeben des Hauptschalters 107.

Patentansprüche

1. Verfahren zur Aktualisierung der Software (20) eines mikrocomputergestützten Telefons (10), das eine Zentralproessoreinheit (101), einen flüchtigen Speicher (104) für die Programmausführung und einen nichtflüchtigen Speicher (103) aufweist, in dem die Software (20) gespeichert ist, wobei die Software (20) mehrere Module (21, 22) aufweist, dadurch gekennzeichnet, daß das Verfahren mittels eines entfernt angeordneten Computersystems (12) über ein Telekommunikationsnetzwerk (11) ausgeführt wird, wobei die in dem nichtflüchtigen Speicher (103) gespeicherte Software (20) des weiteren eine Strukturtabelle (23) mit zumindest einer Versionsnummer bezüglich der Software (20) aufweist, die Software (20) von einem normalen Betriebsmodus in einen Aktualisierungsmodus zum Aktualisieren der Software (20) umschaltbar ist und das Verfahren folgende Verfahrensschritte aufweist:

- Versetzen der Software (20) in einen Aktualisierungsmodus,
- Herstellen einer Telekommunikationsverbindung zu dem Computersystem (12)
- Übertragen der Strukturtabelle (23) an das Computersystem (12),
- Erstellen und Übertragen eines Aktualisierungspakets mit neuen Modulen und einer neuen Strukturtabelle entsprechend der empfangenen Strukturtabelle (23) durch das Computersystem (12) und
- Aktualisieren der vorhandenen Module (21, 22) und der Strukturtabelle (23) der Software (20) mittels der neuen Module und der neuen Strukturtabelle aus dem Aktualisierungspaket.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß jeder Modul (21, 22) der Software (20) in einem vorbestimmten Speicherschlitze (31) des nichtflüchtigen Speichers (103) mit zusätzlichem Speicherraum für die spätere Aktualisierung gespeichert und jeder in Betracht kommende Modul dadurch aktualisiert wird, daß in demselben Speicherschlitze (31) ein entsprechender neuer Modul abgespeichert wird.

3. Verfahren nach Anspruch 2, dadurch gekennzeichnet, daß die Strukturtabelle (23) eine Versionsnummer für einen jeden Modul (21, 22) der Software (20) enthält.

4. Verfahren nach Anspruch 2 oder 3, dadurch gekennzeichnet, daß die Strukturtabelle (23) des weiteren eine Adressentabelle zum Aufzeichnen der Adressen der vorbestimmten Speicherschlitze (31) aufweist.

5. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß die Struktur-

tabelle (23) des weiteren Hardwarestrukturdaten des Telefons (10) enthält.

6. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß das Aktualisierungspaket vor dem Aktualisierungsschritt in dem flüchtigen Speicher (104) zwischengespeichert wird. 5

7. Verfahren nach Anspruch 6, dadurch gekennzeichnet, daß das Telefon (10) einen von der Zentralprozessoreinheit (101) steuerbaren Hauptschalter (107) aufweist, der während des Aktualisierungsschritts gesperrt wird, um eine Unterbrechung des Aktualisierungsschritts durch den Benutzer zu verhindern. 10

8. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß es des weiteren einen Verfahrensschritt zum Eintasten eines Benutzeroptionscodes vor dem Übertragungsschritt aufweist, der es dem Benutzer ermöglicht, eine Aktualisierungsoption einzugeben. 15 20

9. Verfahren nach Anspruch 8, dadurch gekennzeichnet, daß der Benutzeroptionscode nach Zustandekommen der Telekommunikationsverbindung zusammen mit der Strukturtable (23) an das Computersystem (12) übertragen wird. 25

10. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß das Telefon (10) ein Mobiltelefon und das Telekommunikationsnetz (11) ein Mobiltelefonnetz ist.

11. Verfahren nach Anspruch 10, dadurch gekennzeichnet, daß das Mobiltelefonnetz ein GSM-Netz ist. 30

Hierzu 3 Seite(n) Zeichnungen

35

40

45

50

55

60

65

- Leerseite -

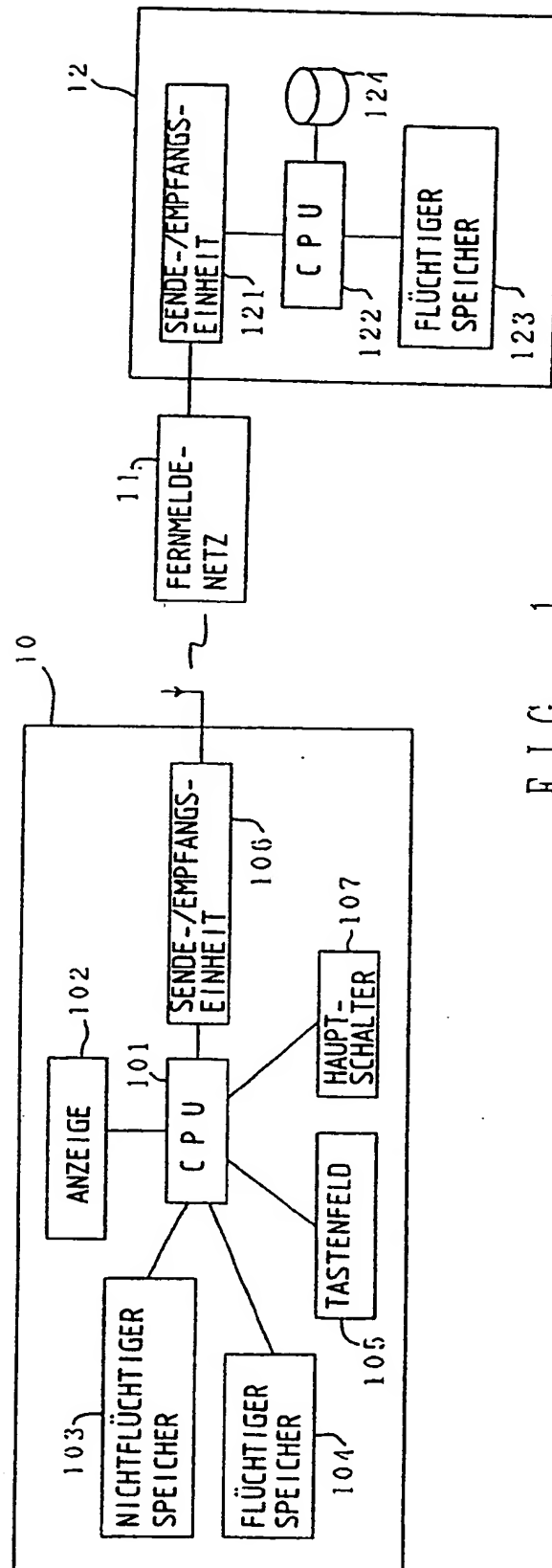


FIG. 1

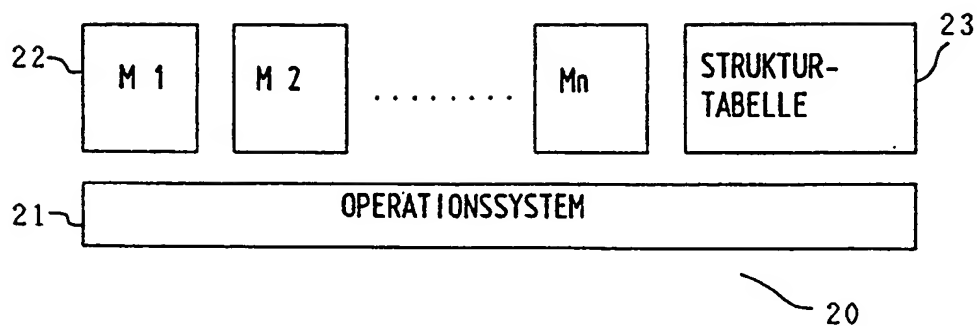


FIG. 2

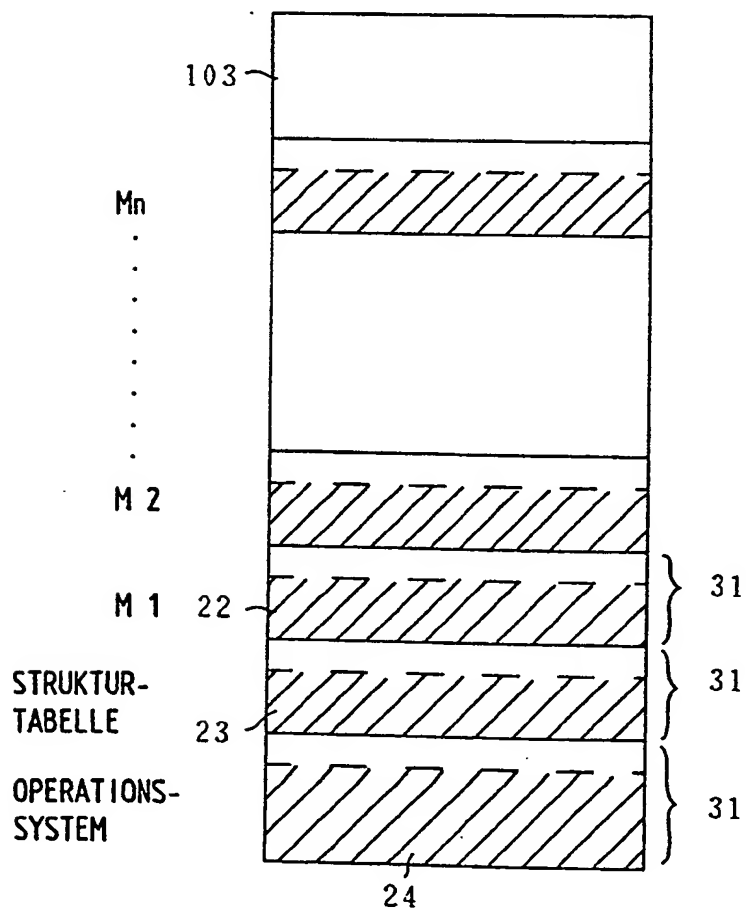


FIG. 3

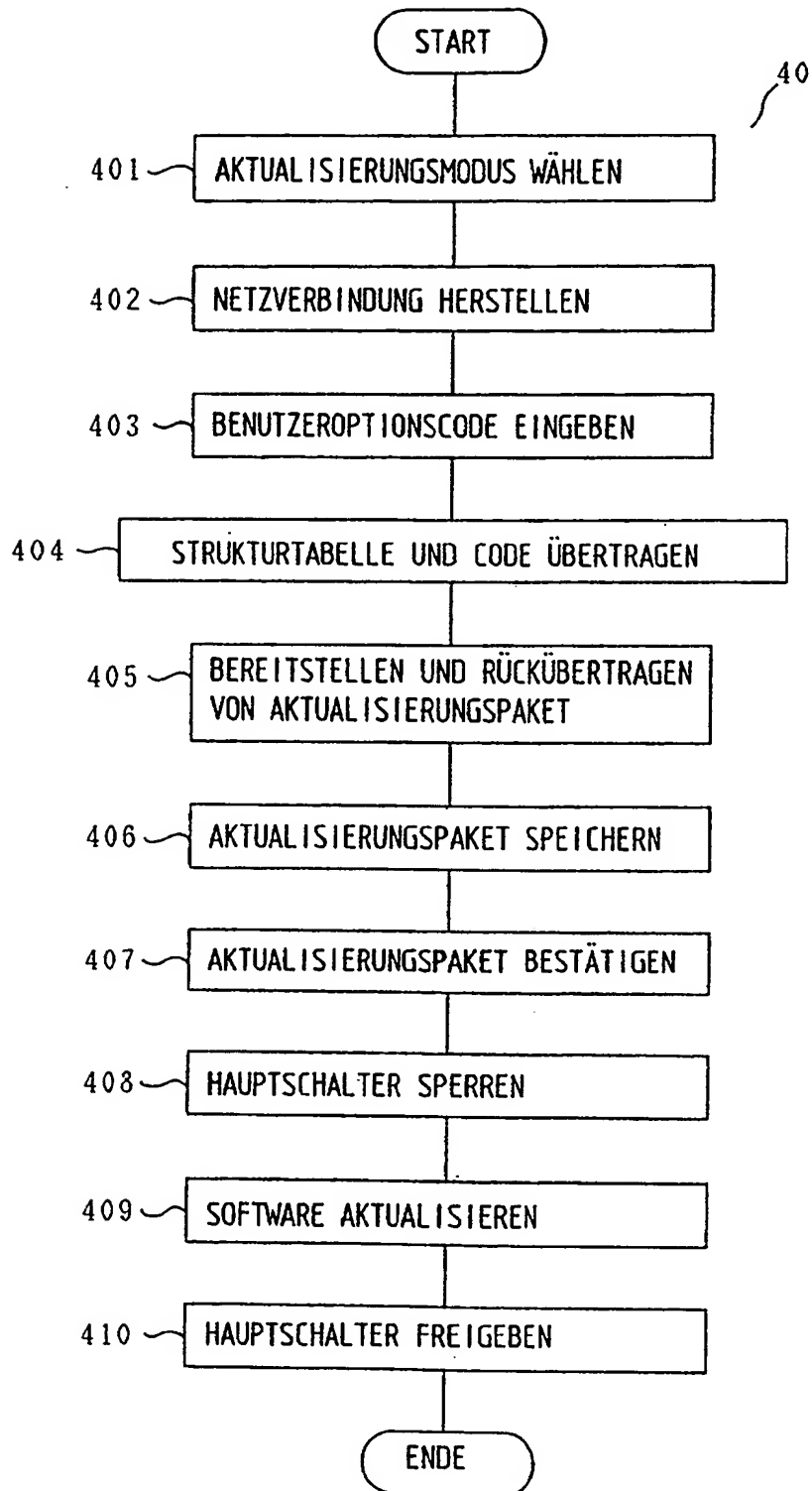


FIG. 4

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.